Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

Лабораторная работа 14

По дисциплине «Основы алгоритмизации и программирования»

На тему «[Многомерные массивы](#Лаб14)»

Выполнила:

Студентка 1 курса 7 группы

Шинкевич Марина Дмитриевна

Преподаватель: асс. Андронова М.В.

2023, Минск

Дополнительные задания к лабораторной 14

1. Дана квадратная матрица порядка **2n**, элементы которой формируются случайным образом и находятся в пределах от −10 до 10. Получить новую матрицу, переставляя ее блоки размера **n×n** в соответствии со схемой.

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "rus");

const int n = 2;

int A[n][n], C[n][n];

srand(time(NULL));//Инициализируем генератор случайных чисел

for (int i = 0; i < n; i++)//Основной цикл для заполнения массива A случайными числами.

{

for (int j = 0; j < n; j++)

{

A[i][j] = rand() % 11 ; //Заполняем элемент массива случайным числом от 0 до 10.

}

}

cout << "Начальная матрица: " << endl;

for (int i = 0; i < n; i++) //Цикл для вывода начальной матрицы A.

{

for (int j = 0; j < n; j++)

{

cout << A[i][j] << " ";

}

cout << endl;

}

for (int i = 0; i < n; i += 2) { // Цикл для перестановки элементов в матрице A и записи их в матрицу C.

for (int j = 0; j < n; j += 2)

{//// Перестановка элементов матрицы A в матрицу C.

C[i][j] = A[i + 1][j + 1];

C[i + 1][j] = A[i][j + 1];

C[i + 1][j + 1] = A[i][j];

C[i][j + 1] = A[i + 1][j];

}

}

cout << "Конечная матрица: " << endl;

for (int i = 0; i < n; i++) //Цикл для вывода конечной матрицы C.

{

for (int j = 0; j < n; j++)

{

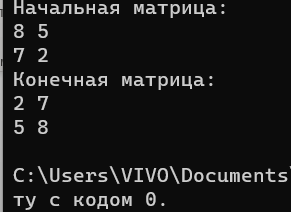
cout << C[i][j] << " ";

}

cout << endl;

}

}



2) Латинским квадратом порядка **n** называется квадратная таблица размером **nхn**, каждая строка и каждый столбец которой содержат все числа от 1 до **n**. Для заданного **n** в матрице **L(n, n**) построить латинский квадрат порядка **n**.

#include <iostream>

using namespace std;

int main()

{

const int n = 10;

int p;

int A[n][n];

for (int i = 0; i < n; i++)

{

p = i + 1;//присваивание переменной p значения i + 1. Переменная p будет использоваться для заполнения элементов массива.

for (int j = 0; j < n; j++) //вложенный цикл, который перебирает значения и используется для заполнения строк массива A.

{

A[i][j] = p;

p++;

if (p > n) //условный оператор, который проверяет, что значение p не превышает n, и если превышает, то присваивает ему значение 1.

p = 1;

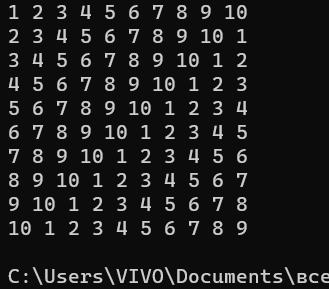
cout << A[i][j] << " ";

}

cout << endl;

}

}



3) Путем перестановки элементов квадратной вещественной матрицы добиться того, чтобы ее максимальный элемент находился в левом верхнем углу, следующий по величине − в позиции (2, 2), следующий − в позиции (3, 3) и т. д., заполнив таким образом всю главную диагональ.

#include <iostream>

#include <iomanip>

using namespace std;

int main()

{

setlocale(LC\_ALL, "rus");

const int n = 4; // Размерность квадратной матрицы.

int A[n][n] = {

{1, 2, 3,4},

{4, 5, 6,1},

{7, 8, 9,10},

{11, 12, 10,13}

};

int max[n];//Создание массива max с размером n для хранения максимальных значений каждой строки массива A.

for (int a = 0; a < n; a++)

{

max[a] = A[a][a];//записывает значение на главной диагонали текущей строки массива A в элемент max[a]. Это гарантирует, что в начале каждой итерации максимальное значение для этой строки будет равно значению на главной диагонали.

for (int i = 0; i < n; i++)//цикл , который проходит по каждой строке массива A.

{

for (int b = 0; b < n; b++)//Внутри вложенного цикла есть еще один вложенный цикл , который проходит по каждому элементу в строке массива A.

{

if (A[i][b] > max[a]) // если текущий элемент A[i][b] больше текущего максимального значения max[a] для этой строки

{

bool bul = true;

for (int m = 0; m < n; m++)//проверяется, не совпадает ли текущий элемент max[m] с текущим элементом A[i][b]. Если они совпадают, переменная bul становится false, иначе она остается true

{

if (max[m] == A[i][b]) {

bul = false;

}

}

if (bul)//Если bul равно true, значит, текущий элемент A[i][b] является новым максимальным значением для этой строки, и происходит обмен максимального значения max[a] с текущим элементом A[i][b] при помощи функции swap. Таким образом, максимальное значение для этой строки обновляется на новое максимальное значение.

{

max[a] = A[i][b];

swap(A[i][b], A[a][a]);

}

}

}

}

}

for (int i = 0; i < n; i++)//вывод результата

{

for (int b = 0; b < n; b++)

{

cout << A[i][b] << ' ';

}

cout << '\n';

}

return 0;

}

